A single machine scheduling problem with bin packing constraints

Jean-Charles Billaut, Federico Della Croce, Andrea Grosso

A single machine scheduling problem with bin packing constraints

Jean-Charles BILLAUT[1], Federico DELLA CROCE[2] and Andrea GROSSO[2]

[1] L.I.,Université François-Rabelais Tours, France jean-charles.billaut@univ-tours.fr
[2] D.A.I., Politecnico di Torino, Italy federico.dellacroce@polito.it
[3] D.I., Universitá di Torino, Italy grosso@di.unito.it

Keywords: Scheduling, Bin packing, Heuristics.

We consider a single machine scheduling problem with additional bin packing constraints. The origin of the problem comes from the production of chemotherapy drugs (Mazier *et. al.* 2010). In this production environment, raw materials are called monoclonal antibodies and can be stored in vials for a long time before use (Billaut 2011). However, once a vial is opened or once the active agent has to be mixed with some water, it must be used before a given time limit, in order to keep intact the properties of anticancer active agents. The maximum delay of use after opening depends on the agent and may vary between several hours to several days. In the mean time, the product has to be stored in a fridge for temperature and darkness reasons. The cost of these drugs is not negligible and the economic impact of saving these products is very important. The preparations that are scheduled and which use the same vial have to be completed before the product perishes. In other words, the total processing time of the jobs assigned to a same vial cannot be greater than the life time of the raw material. Furthermore, the total consumption cannot be greater than the total volume of the vial. Because each preparation has to be delivered to a patient for a given due date, one objective of the problem is to minimize the maximum lateness related to these due dates. Notice that in such a context, the deadline for the use of the raw material becomes a variable of the problem, which is directly related to the production scheduling decisions: once the life duration or the capacity of the vial is exceeded, a new vial is opened.

1 Problem statement and notations

We consider a simplified version of the above problem, with only one machine and one type of raw material. We consider a set of n jobs to schedule on a single machine. W.l.o.g., the jobs are supposed to be numbered in EDD order, i.e. $d_1 \leq d_2 \leq ... \leq d_n$. To each job $j \in \{1, ..., n\}$ is associated a processing time p_j, a consumption b_j and a due date d_j. The life duration of the product after opening is equal to T and the volume of one vial is equal to V. We assume always w.l.o.g. that $p_j < T$ and $b_j < V$, $\forall j$, $1 \leq j \leq n$. The number of vials is not limited but supposed to be bounded by n. We denote by C_j the completion time of j, L_j the lateness defined by $L_j = C_j - d_j$. The maximum lateness is defined by $L_{max} = \max_{1 \leq j \leq n} L_j$. We assume that the maximum lateness is bounded by Q. Minimizing the quantity of lost raw materials is equivalent to minimize the number of vials that are opened. Therefore, the problem is a mixed between a scheduling problem and a two-constraint bin packing problem. Without due dates (or with extremely large due dates), the problem is a bin packing problem. For this reason, the problem is clearly NP-hard. With a huge T and a huge V, the problem is the trivial single machine problem with the L_{max} minimization. In the following, we call a "bin", the set of jobs performed with the same vial.

Example: We consider a set of six jobs with the following data and $T = 10$ and $V = 10$.

j	1	2	3	4	5	6
p_j	3	4	4	5	3	1
b_j	1	2	5	3	1	4
d_j	7	9	11	13	14	16

The schedule $(1,3,5,2,4,6)$ is represented in Fig. 1. In this two dimensional Gantt chart, a job j is represented by a rectangle with the duration p_j on the x-axis and the consumption b_j on the y-axis. Jobs of the same bin are connected by the north-west corner of the rectangle. The first job of a bin is put on the x-axis. In Fig. 1, one can see that 1 and 2 are the first jobs of the bins $(1,3,5)$ and $(2,4,6)$. The maximum lateness of this sequence is equal to $L_{max} = \max(-4,5,-4,6,-4,4) = 6$ and this schedule requires 2 bins.

We denote by $u_k \in \{0,1\}$ a boolean variable equal to 1 if bin k is used, and 0 otherwise and by $x_{j,k} \in \{0,1\}$ a boolean variable equal to 1 if job j is assigned to bin k, and 0 otherwise. The problem can be formalized as follows.

$$\text{MIN} \sum_{k=1}^{n} u_k$$

s.t.

$$\begin{cases} \sum_{k=1}^{n} x_{j,k} = 1, \forall j \in \{1,...,n\} & (1) \\ \sum_{j=1}^{n} p_j x_{j,k} \leq T u_k, \forall k \in \{1,...,n\} & (2) \\ \sum_{j=1}^{n} b_j x_{j,k} \leq V u_k, \forall k \in \{1,...,n\} & (3) \\ \sum_{h=1}^{k-1} \sum_{i=1}^{n} p_i x_{i,h} + \sum_{i=1}^{j} p_i x_{i,k} \leq \sum_{i=1}^{n} (d_i + Q) x_{i,k} + HV(1 - x_{j,k}), & \\ \qquad \forall j \in \{1,...,n\}, \forall k \in \{1,...,n\} & (4) \\ u_{k+1} \leq u_k, \forall k \in \{1,...,n\} & (5) \end{cases}$$

Constraints (1) ensure that each job is performed by using one vial (is assigned to one bin). Constraints (2) and (3) correspond to the temporal and capacity limits. Constraints (4) suppose that job j is in bin k and correspond to the bound on the L_{max}: $\sum_{h=1}^{k-1} \sum_{i=1}^{n} p_i x_{i,h}$ is the completion time of the $k-1^{th}$ bin, $\sum_{i=1}^{j} p_i x_{i,k}$ is the completion time of job j in its bin (remember that the jobs are numbered in EDD order). Constraints (5) ensure that the bins are used in their index increasing order.

2 Solution approaches

2.1 Recovering beam search algorithm

The *Beam Search* algorithm is a truncated branch-and-bound method where a subset of w nodes at each level are selected for branching. w is called the *beam width.* This method was first proposed in (Ow and Morton 1988). For the selection of nodes, each node is evaluated by a combination of a lower bound (LB) and an upper bound (UB), generally a weighted sum $V = (1-\alpha)LB + \alpha UB$. Because the selected nodes are not necessarily the bests at a given level of the tree, among the set of possible nodes of a pure branch-and-bound algorithm, a recovering phase is applied in the *Recovering Beam Search* algorithm (RBS). The aim of this phase is to recover from wrong decisions jumping to a better node at the same level of the search tree. For a detailed description of RBS we refer to (Della Croce *et. al.* 2004). A node σ of the tree is defined by a partial sequence of jobs $S(\sigma)$, a list of unscheduled jobs $\overline{S}(\sigma)$, a lower bound $LB(\sigma)$, and an upper bound $UB(\sigma)$. At the

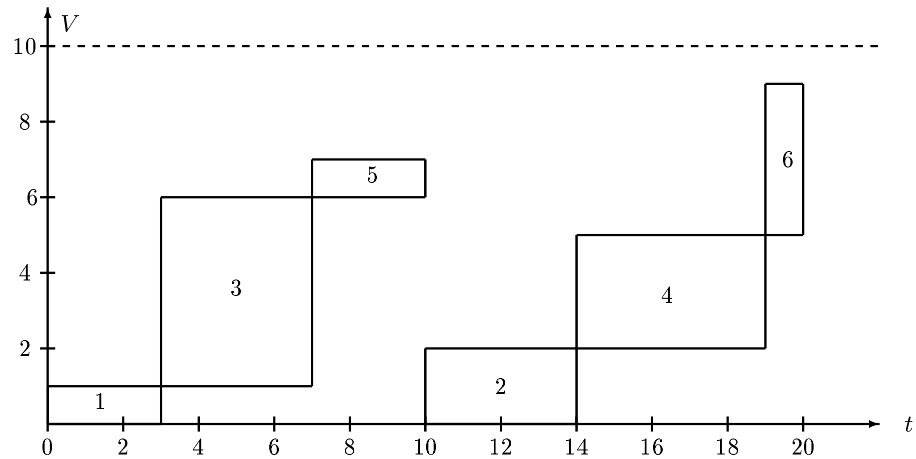

Fig. 1. Gantt chart of schedule $(1, 3, 5, 2, 4, 6)$

root node, the initial sequence of unscheduled jobs is determined as follows. Starting from EDD sequence, a steepest descent algorithm is used to reduce the number of bins, without violating the constraint on the L_{max}. The initial sequence which is obtained is called INIT. Alg. 1 and Alg. 2 describe the lower bound and upper bound computation. We denote by $Bin(S)$ a function that computes in $O(n)$ the number of bins used by a sequence S and the sum of jobs processing time and of jobs consumption in the last bin (respectively called $RestP$ and $RestB$). $u//v$ stands for the concatenation of u and v.

Algorithm 1 $LB(\sigma)$

$NbBins = Bin(S(\sigma)) - 1$
$SumP = RestP + \sum_{j \in \overline{S}(\sigma)} p_j$
$SumB = RestB + \sum_{j \in \overline{S}(\sigma)} b_j$
$NbBins = NbBins + \max(\lceil SumP/T \rceil, \lceil SumB/V \rceil)$
Return($NbBins$)

Algorithm 2 $UB(\sigma)$

$S'(\sigma) = S(\sigma)//\overline{S}(\sigma)$
$NbBins = Bin(S'(\sigma))$
Return($NbBins$)

The evaluation of a node is given by $V(\sigma) = (1 - \alpha)LB(\sigma) + \alpha UB(\sigma)$. The recovering phase is composed by two types of neighborhood called SWAP and EBSR (extraction and backward shift reinsertion) proposed in (Della Croce *et. al. 1* 2004).

2.2 Matheuristic algorithm

A matheuristic procedure (Della Croce *et. al.* 2013) can be seen as a local search approach for MIPs, expecially suited for $0 - 1$ variables, using a generalization of the k-exchange neighborhood. Consider a general MIP ($\min c^T X$ subject to $AX \leq b$, $X \in \{0, 1\}$), where $X^T = (x_1, x_2, .., x_n)$ is a vector of n variables of the problem and $\bar{X}^T =$

$(\bar{x_1}, \bar{x_2}, .., \bar{x_n})$ is a feasible solution to the MIP. If this is the case, it is always possible to define a subset S of a defined size of variables indices $\{1, 2, .., n\}$. The neighborhood $N(\bar{X})$ consists of all solutions of the MIP where the j^{th} variable is equal to the value of the j^{th} variable in $\bar{X}$ for all $j \notin S$, namely $N(\bar{X}) = \{X \mid x_j = \bar{x}_j,\ \forall j \notin S\}$. The resulting neighborhoods $N(\bar{X})$ can then be searched for an improving solution using a MIP-solver both optimally or approximately. The main idea stems in representing the MIP as a permutation problem where variables belonging to the current solution are partitioned into two sets. A first set $(\bar{X}/S)$ is then reoptimized by means of a MILP solver generating a permutated assignment while variables in the second set (S) keep the same assignment as in the current solution. For the considered problem, the incumbent solution returned by the RBS algorithm induces correspondingly a sequence of the bins where we assume that γ bins are used. The neighborhood exploration works as follows. Starting with the first bin, consider the r-th bin in the sequence along with bins $r+1$, $r+2$,..., $r+H-1$ with item set $S = S_r \cup S_{r+1} \cup ... \cup S_{r+H-1}$. Solve the problem of rescheduling the items in S so that $w(S_{r+H-1}) = \sum w_i : i \in S_{r+H-1}$ is minimized where we use $w_i = \max\{v_i, p_i\}$ (this performance measure has been selected experimentally). The rationale is to empty as much as possible bin $r+H-1$. If a (sub)sequence with $w(S_{r+H-1}) = 0$ is obtained, then one bin is saved, γ is reduced by one unit and the process can restart with $r = 1$. Alternatively, the new subsequence is kept anyway as the space of bin S_{r+H-1} has been optimized and will be used in the next iterate. The approach is then iterated for $r = 1, 2, ..., \gamma - H + 1$. Whenever $r = \gamma - H + 1$ is reached, the process restarts with $r = 1$ until a time limit is exceeded. The problem of rescheduling the items in S is done by solving by means of an ILP solver that adapts the model presented to above in order to take into account the fact that bins $1, ...r-1$ and $r+H, ..., \gamma$ are not rescheduled and that the objective is to minimize the weight of the items assigned to bin $r+H-1$.

3 Computational experiments

Detailed computational experiments will be presented at the Conference and relate to the bin packing instances considered in (Capara and Toth 2001) modified in order to take into account also the sequencing part of the problem. Here, we simply stress that the proposed approach combining RBS and MH reaches in limited time good quality solutions being able to improve also some of the benchmarks results on the original bin packing instances of (Capara and Toth 2001).

References

J-C Billaut, 2011, "New scheduling problems with perishable raw materials constraints", *16th IEEE Conference on Emerging Technologies and Factory Automation*, Vol. 01, pp. 1-5.

A. Caprara and P. Toth, 2001, "Lower bounds and algorithms for the 2-dimensional vector packing problem", *Discrete Applied Mathematics*, Vol. 111, pp. 231-262.

F. Della Croce, M. Ghirardi and R. Tadei, 2004, "Recovering Beam Search: enhancing the beam search approach for combinatorial optimization problems", *Journal of Heuristics*, Vol. 10, pp. 89-104.

F. Della Croce, A. Grosso and F. Salassa, 2013. "Matheuristics: Embedding MILP Solvers into Heuristic Algorithms for Combinatorial Optimization Problems", in Heuristics, Theory and Applications, P. Siarry (ed.), Nova Publishers, 31-52.

A. Mazier, J-C. Billaut and J-F. Tournamille, 2010. "Scheduling preparation of doses for a chemotherapy service", *Annals of Operations Research*, Vol. 178(1), pp. 145-154.

P. S. Ow and T. E. Morton, 1988,"Filtered beam search in scheduling", *International Journal of Production Research*, Vol 26, pp. 297–307.